

End User Development in Small and Medium-Sized Enterprises: Empirical Findings Concerning the Adaptation Process of Software Systems

Christian Dörner, Volkmar Pipek, Matthias Betz, Jan Heß,
Markus Rohde, Volker Wulf

Chair of Information Systems and New Media
University of Siegen
Hölderlinstraße 3
57068 Siegen

{christian.doerner, volkmar.pipek, matthias.betz, jan.hess,
markus.rohde, volker.wulf}@uni-siegen.de

Abstract: This article describes the need of small and medium-sized enterprises for adaptation. We present some concepts of end user development that could help companies realize the adaptation of their software systems. Moreover, we outline all persons who are involved in the adaptation process in these companies and how the division of labour is realized. We support our thesis with the empirical findings of one of our studies.

1 Introduction

The development of appropriate software systems has been a research topic in the area of computer science for many years. Nowadays end users participate more than ever in the development of software systems. This is valid both for the participation at design time and the adaptation of the systems at use time. The research area *end user development* intends to implement the “easy-to-develop” principle for software systems [LPW05]. The implementation of this principle will become more complex in case of a “casual user” (one without any special knowledge in computer science) adapting and developing software systems on his own [SM04]. This article focuses on end user development of software systems in small and medium sized-enterprises (SME¹). Similar to larger companies, SME tend to use computer systems in order to plan and execute business processes. However, in contrary to these companies, the market demands SME to be highly flexible and to adjust their processes properly, with a limited amount of human resources. The problem is that many SME don’t even have an own IT-department. Therefore, they struggle with many changes and must often adjust their division of labour. The software systems of the SME do not match this flexibility. The companies usually run standard software products because the costs of these products are much lower than the costs of individual software. The need of flexibility and the use of standard software leads to a gap. In order to bridge

¹In contrary to the definition of the European Union we define SME as companies with a limit of 500 employees.

this gap, SME would need highly adaptable standard software that fits to their needs and could be integrated into their existing infrastructure. The management of the infrastructure is very often done by the end users themselves, due to the lack of expertise in this field in SME. Putting this together, it becomes clear that the concepts of end user development are very important for the SME.

The next chapter focuses on selected concepts of end user development, which are important for the development of highly adaptable software systems for SME.

2 Concepts of End User Development

There are many concepts for the development of highly adaptable software systems in the literature. Paternó [Pat99], for example, presents an approach that is interesting for the developmental task of highly adaptable systems for SME. Others, like Lieberman et al. [LPW05] give a more holistic view on the topic. The following introduces three concepts, which are significant to the development of adaptable systems on the architectural, the interface, and the socio-technical level.

Adaptation features Mørch [Mør97] and Fischer and Girgensohn [FG90] suggest different forms of *adaptation features* as a design approach in order to close the gap between the use context of a system and its underlying technological structures. It aims to create a gentle slope of complexity between simple but restricted modifications and powerful but complex modifications by architectural means. Such a construction of the system would ensure that many end users have the option to adapt the system on a level of complexity they understand.

Software visualization The term software visualization refers to the visual programming concept. Green and Petre [GP96] analysed programming environments and attested that visual programming concepts are of major importance. The same applies to the use of visual programming concepts that are used for the adaptation of SME software systems. The employees don't need to learn a programming language in order to make the adjustments. They can modify their systems in a natural way, focussing on their tasks and not on the technology they need to use for the adaptation. An appropriate visual programming language can make use of the signs the users are familiar with, and it will hide the complexity of the system by clustering and blackboxing functionality.

Socio-technical support for appropriation This concept enables end users to collaborate when adjusting their infrastructure. The collaboration support could follow the vision of "virtual user communities of practice", and enable participants to share and exchange their knowledge, when making changes to the system and even allow them to share already adapted parts of the systems. A detailed discussion of different collaborative aspects can be found in *Tailoring as Collaboration* [MM00].

While these measures can be assumed to be interesting for many organisations in many domains, our interest is to understand the requirements for these types of EUD support

for Enterprise Resource Planning (ERP) Systems, because these systems are the most important software systems of SMEs. Therefore, we started a strong empirical effort, which is described in the next section, to understand whether these concepts address the needs felt in different types of SME.

3 Empirical Findings

This contribution covers only the first phase of our empirical analysis. We made semi-structured, explorative interviews with seven persons of five different SME, which do business in various industries. Almost all interviewees were technically experienced and worked in the IT-departments of the companies. In case of the smaller companies we interviewed the CEO (Chief Executive Officer). Two of the five companies had a maximum of 30 employees, while the other three companies had a maximum of 200 employees. The interviews aimed to understand how size matters, and what different types of roles were involved in adaptation processes.

Playing Field All companies had a heterogeneous software infrastructure that was dominated by professional ERP or ERP-like Systems, which were complemented with various standard software products, like Office Tools, but also with individual software. The individual software served special domain-dependent purposes or connected two standard applications with each other.

Participants, Roles and Competencies in the Adaptation Process In the small companies, not all staff members had frequent contact with the software infrastructure, but if they had, they had quite diverse competencies regarding use and management of the software infrastructure. The CEO was usually responsible for infrastructure decisions, but some decisions were also delegated to staff members, based on their interest and experience. The competencies were usually developed when facing an infrastructure breakdown. The external actors who were involved in infrastructure decisions came from the technological domain (e.g. local hardware store), but as well from the management domain (e.g. tax consultant). Competing businesses in the same service sector did not have an influence, but collaborating partners occasionally communicated ideas for infrastructure improvement. With the bigger companies, the roles involved in infrastructure development were surprisingly not much more differentiated. IT departments were very small (2-5 persons) and – in their everyday routines – focused technically keeping the infrastructure alive. They were only adapting their systems when they either experienced more severe problems with them, when the release policy of the software provider forced them to, or when they had an urgent need to adapt the system to changing business processes. Even then, adaptation only took place if the connected costs were acceptable, which resulted in quite few adaptations. The adaptation problems that the SME were able to address by themselves were rather minor. All major problems, such as the connection of the different software systems or the adaptation of their ERP systems, were solved by external consultants, who were naturally more costly for the company. The smaller companies were more dependent on informal social networks to get help for their infrastructure problems, while the bigger

companies addressed the problem in a more defined way. Still, the frequently changing division of labour within the SME that depends on workload, interest and adaptation goals, showed that it is very likely that the participants in the adaptation process will change quite often.

4 Summary/Conclusion

In this article we described the relevance of end user development for SME. This relevance roots in the discrepancy between the need of SME for continuous adaptation of their software infrastructure and their use of standard software. We identified three important strategies to provide EUD support on the architectural, the interface and the socio-technical level. We described preliminary findings from an ongoing empirical study of five SME that helps to illustrate the relevance of these support approaches for SME. In general, the cost structure of infrastructure maintenance indicates a huge impact of EUD approaches for ERP systems. We have to see, what latest concepts like service-oriented infrastructures can offer on the architectural level. On the interface level we may have to address end users with very diverse backgrounds (at least in the bigger companies), which may lead to problems for domain-oriented approaches due to a lack of a shared application domain. The socio-technical level seems to be the most interesting one, since in all companies infrastructure maintenance was assumed to be a collaborative task of many internal and external participants. The constantly changing involvement and shift of responsibilities of the participants may be the biggest challenge here, due to the tight consideration of productive and infrastructural parts of the work .

References

- [FG90] G. Fischer and A. Girgensohn. End-User Modifiability in Design Environments. In *Proceedings of CHI '90*, pages 183–191, Seattle, 1990.
- [GP96] T. R. G. Green and M. Petre. Usability analysis of visual programming environments: a “cognitive dimensions” framework. *Journal of Visual Languages and Computing*, 7(2):131–174, 1996.
- [LPW05] Henry Lieberman, Fabio Paternò, and Volker Wulf. *End User Development*. Springer Netherland, Berlin, 1 edition, 2005.
- [MM00] A. I. Mørch and N. D. Mehandjiev. Tailoring as Collaboration: The Mediating Role of Multiple Representations and Application Units. *Computer Supported Cooperative Work*, 9(1):75–100, 2000.
- [Mør97] A. Mørch. *Three Levels of End-user Tailoring: Customization, Integration and Extension*, pages 51–76. The MIT Press, Cambridge, MA, 1997.
- [Pat99] F. Paternò. *Model-based Design and Evaluation of Interactive Applications*. Springer Verlag, Berlin, 1999.
- [SM04] A. Sutcliffe and N. Mehandjiev. End User Development. *Special Issue of the Communications of the ACM*, 47(9):31–32, 2004.